NASA-CR-190651

DEVELOPMENT OF ITERATIVE TECHNIQUES FOR THE SOLUTION OF UNSTEADY COMPRESSIBLE VISCOUS FLOWS

LANGLEY GRANT
IN-34-CR
115128
P.30

Grant NAG-1-1217

Progress Report for the Period

February 14, 1992 - August 13, 1992

Submitted to

NASA Langley Research Center
Hampton, VA 23665

Attn: Dr. Woodrow Whitlow

Prepared by

Lakshmi N. Sankar
Professor, School of Aerospace Engineering

Duane Hixon
Graduate Research Assistant

School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332

August 1992

(NASA-CR-190651) DEVELOPMENT OF ITERATIVE TECHNIQUES FOR THE SOLUTION OF UNSTEADY COMPRESSIBLE VISCOUS FLOWS Semiannual Progress Report, 14 Feb. - 13 Aug. 1992 (Georgia Inst. of Tech.) 30 p

N92-34237

Unclas

G3/34 0115128

I. Introduction

During the past two decades, there has been significant progress in the field of numerical simulation of unsteady compressible viscous flows. At present, a variety of solution techniques exist such as the transonic small disturbance analyses (TSD) [e.g. Ref. 1-3], transonic full potential equation-based methods [e.g. Ref. 4-6], unsteady Euler solvers [e.g. Ref. 7-8], and unsteady Navier-Stokes solvers [e.g. Ref. 9-12]. These advances have been made possible by developments in three areas: (1) Improved numerical algorithms, (2) Automation of body-fitted grid generation schemes, and (3) Advanced computer architectures with vector processing and massively parallel processing features.

Despite these advances, numerical simulation of unsteady viscous flows still remains a computationally intensive problem, even in two dimensions. For example, the problem of dynamic stall of an oscillating NACA 0012 airfoil using state of the art alternating direction implicit (ADI) procedures presently require between 10,000 and 20,000 time steps per cycle of oscillation at low reduced frequencies when the viscous flow region is sufficiently resolved [Ref. 9]. In three dimensions, unsteady Navier-Stokes simulations of a helicopter rotor blade in forward flight requires over 30,000 time steps or more for a full revolution of the rotor [Ref. 10]. In other unsteady flows, such as the high angle of attack flow past fighter aircraft configurations, a systematic parametric study of the flow is presently not practical due to the very large CPU time needed for the simulations [Ref. 13]. Thus, it is clear that significant improvements to the existing algorithms, or dramatic improvements in computer architectures will be needed, before unsteady viscous flow analyses become practical day-to-day engineering tools.

One scheme that has been of recent interest is the Generalized Minimal RESidual (GMRES) method originally proposed by Saad and Schultz (Ref. 14). This procedure uses a conjugate gradient method to accelerate the convergence of existing flow solvers. GMRES was added to existing steady flow solvers by Wigton, Yu, and Young (Ref. 15), and to an unstructured grid

flow solver by Venkatakrishnan and Mavriplis (Ref. 16). Saad has also used a Krylov subspace projection method on a steady, incompressible Navier-Stokes problem and an unsteady one dimensional wave propagation equation (Ref. 17).

Under NASA Langley support, a research effort was initiated at Georgia Tech in February 1991 on the development of efficient techniques for the computation of 2-D and 3-D unsteady compressible flow problems. It was found that in 2-D unsteady viscous flow applications, the GMRES scheme was able to significantly improve the accuracy and stability characteristics of an existing 2-D ADI (Alternating Direction Implicit) time marching scheme. That is, the GMRES/ADI combination allowed 10 to 20 times larger time steps compared to an ADI scheme. Because the GMRES algorithm requires 5 to 10 times the CPU work compared to the ADI scheme, the combined GMRES/ADI scheme yields a net factor of 2 savings in CPU cost.

During the past year, we also experimented with a GMRES/multigrid/ADI combination. The purpose of this combination was to compute the low frequency components of the change in the flow properties from one time step to the next on a coarse grid. This strategy reduces the memory requirements of the GMRES method roughly by a factor of 4-8 for steady flow problems.

These findings have been documented in the AIAA Paper 92-0422 by Hixon and Sankar, presented in Reno, and also in our previous two progress reports.

Progress During the Reporting Period

During the present reporting period (February 1992-August 1992), our emphasis shifted toward 3-D simulations. We modified an existing 3-D ADI Navier-Stokes solver into a GMRES/ADI solver. For validation of the flow solver, we have selected the following test cases:

(a) Steady transonic flow past an F-5 wing.

(b) Unsteady transonic flow past an F-5 wing with a sinusoidally oscillating trailing edge flap.

(c) Deep dynamic stall of a 3-D NACA 0015 rectangular wing.

We have completed sample calculations with the GMRES/ADI solver for cases (a) and (c), and ADI calculations for case (b). Our experiences with the GMRES/ADI procedure in such 3-D applications are discussed below.

i) Experiences with GMRES using ADI preconditioner

The derivations of the hybrid ADI solver and the GMRES solver are given in Appendices A and C, respectively.

Viscous transonic flow over an F-5 wing at zero angle of attack was chosen as the baseline case, due to the extensive experimental data available. The Mach number was 0.9, and the Reynolds number was 11 million. The GMRES/ADI code was tried in the Navier-Stokes mode, and it was found that the GMRES version refused to converge completely regardless of the number of directions used. Instead, the solver would 'hang up' at a given residual level, and never converge beyond it.

This problem had occurred in the past in some of our 2-D transonic flow simulations, and usually meant that more GMRES directions were required. Therefore, a series of runs were tried, varying both the number of directions and the ε parameter, which controls the numerical derivatives; while the rate of initial convergence differed, the final solution was similar (and incorrect). At the residual level reached by the GMRES solver, a shock was predicted that does not exist in the converged ADI solution or the experimental results. The result of a 5 direction GMRES/ADI run is compared to the ADI solution in Figures 1,2 and 3.

These problems were eventually traced to the high frequency spatial oscillations in the correction vectors, and were fixed as discussed under heading (iv).

ii) dynamic stall workshop

Carina Tan invited us to a dynamic stall workshop at the NASA Ames Research Center. This workshop was designed to illustrate the state of the art

in unsteady viscous flow predictions. A variety of people, each representing different approaches to solving this problem, were invited to compare their solutions to experimental data obtained by Ray Piziali. Ours was one of two 3D CFD solutions presented.

The experiments were performed with a rectangular wing (AR = 5) using a NACA 0015 section. The wing was pitched 4° about mean angles of 11°, 13°, 15°, and 17° mean angles of attack, at frequencies of 4 Hz, 10 Hz, and 14 Hz. Experimental data was provided for all cases except for the 15° case, in order to tune the code. The challenge provided was to compute the 15° runs without knowing the answer beforehand. The experimental results for the 15° case were provided on arrival at the workshop.

Since the GMRES version of the code was not ready, the original hybrid ADI solver was used. It is planned to re-run the short case with GMRES to compare it to this solution. Because 3-D dynamic stall simulations are CPU intensive, a coarse grid (121 x 21 x 41) was used, along with a large time step ($\Delta t = .01$). Even so, the short case (14 Hz) took 8 hours of CPU time, with the longest case (4 Hz) requiring 15 hours on the Cray YMP. Sample results are given in Figures 4, 5, 6, 7,and 8.

For an initial check of the unsteady GMRES solver, a 5 direction run with 20 times the ADI time step was started (this gives roughly a factor of 2 reduction in CPU time compared to the ADI solver). The preliminary results are given in Figure 9. For the attached flow regime, these preliminary results are very encouraging.

iii) formulation and implementation of LU solver

After the workshop, attention was focused on obtaining a steady solution for the F-5 wing from GMRES. It was postulated that the directions generated by the ADI preconditioner contained high frequency spatial oscillations, and a preconditioner giving 'smoother' directions was sought.

The LU-SGS scheme was chosen as the new preconditioner. The formulation of this scheme is given in Appendix B. Upon implementation, it was found that the LU solver did not converge to an acceptable solution, and also predicted a shock in the flow field. At present, it is thought that this could be

an implementation error, and is being rechecked. Sample results are given in Figures 10, 11, and 12.

The GMRES solver with the LU preconditioner, however, was more stable than it was with the original ADI scheme. With the ADI, it was necessary to turn on the turbulence model after a number of iterations in order to keep the solution from blowing up; this is not necessary with the LU preconditioner.

Unfortunately, convergence of the residuals in the GMRES/LU solver stalled, and still predicted the fictitious shocks. Sample results are given in Figures 13, 14, and 15.

iv) Effects of increasing implicit dissipation on ADI solver

As stated earlier, the GMRES/ADI scheme stalled after just a few iterations. The weighting coefficients by which the correction vectors are multiplied did not converge to zero as the number of directions increased. In fact, these weights were oscillatory, changing sign. This indicated a 'Gibbs'-like phenomenon, where the higher direction vectors attempt to correct (with a negative weight) the errors in the lower direction vectors.

It was postulated that the first few directions from the GMRES solver contained high frequency spatial oscillations, and were noisy (a carpet plot of some earlier 2-D solutions indicated such a behavior in 2-D transonic flow). Thus, these high frequency oscillations must be filtered out before the components are added to the flow properties at $q^{n+1,k}$ (at iteration level 'k') to get $q^{n+1,k+1}$. In the present approach, such a filtering out may be done either as a separate post-processing of the quantity $q^{n+1,k+1} - q^{n+1,k}$, or through implicit smoothing. The latter is easier, and requires increasing the implicit dissipation coefficient ϵ_I, which is discussed in Appendix A.

This idea was recently tested by increasing the implicit dissipation on the left hand side of the ADI equations to smooth the residuals for the GMRES routine. The implicit factor was increased from 5 to 20, and run with 5 directions; it was unstable, but the direction coefficients looked much better than usual. The factor was reduced to 10, and the GMRES routine got the correct, shock-free result for the first time. It was an encouraging sign that a 5

direction run converged enough to get this answer; usually 20 or more directions were required for a trustworthy transonic solution in 2D.

Also, a 20 direction run was performed with the implicit factor set to 20. The asymptotic convergence rate is comparable to our best ADI convergence rate. At the early iterations, however, the GMRES scheme is searching for the steepest descent directions, and shows a slow convergence rate. Results of these runs are shown in Figures 16, 17, and 18.

Proposed Work

A multigrid version of the 3D code is under development presently. It is felt that this will speed the GMRES convergence to the steady solution much as the 2D version did.

We are also planning to run two test cases in the unsteady mode with the GMRES/ADI solver: an F-5 wing with an oscillating trailing edge flap, and the 14 Hz 15° mean angle of attack NACA 0015 wing case from the dynamic stall workshop.

References

1. Borland, C.J. and Rizzetta, D., "Nonlinear Transonic Flutter Analysis," AIAA Paper 81-0608-CP, AIAA Dynamic Specialists Conference, 1981.

2. Rizzetta, D.P. and Borland, C., "Numerical Solution of Unsteady Transonic Flow over Wings with Viscous-Inviscid Interaction", AIAA Paper 82-0352, January 1982.

3. Batina, J. T., "Unsteady Transonic Algorithm Improvements for Realistic Aircraft Applications", AIAA Paper 88-0105, January 1988.

4. Sankar, L.N., Malone, J.B., and Tassa, Y., "An Implicit Conservative Algorithm for Steady and Unsteady Three-Dimensional Transonic Potential Flows", AIAA Paper 81-1016-CP, June 1981.

5. Malone, J.B. and Sankar, L.N., "Application of a Three-Dimensional Steady and Unsteady Full Potential Method for Transonic Flow Computations", AFWAL-TR-84-3011, Flight Dynamics Laboratory, Wright Patterson Air Force Base, Dayton, Ohio, 1984.

6. Shankar, V., Ide, H., Gorski, J. and Osher, S., "A Fast, Time-Accurate Unsteady Full Potential Scheme", AIAA Paper 85-1512-CP, July 1985.

7. Pulliam, T.H., and Steger, J.L., "Implicit Finite Difference Simulations of Three-Dimensional Compressible Flow", AIAA Journal, Vol. 18, 1980.

8. Batina, J.T., "Unsteady Euler Solutions Using Unstructured Dynamic Meshes", AIAA Paper No. 89-0115, January 1989.

9. Sankar, L. N. and Tang, W., "Numerical Solution of Unsteady Viscous Flow past Rotor Sections", AIAA Paper 85-0129.

10. Wake, B. E. and Sankar, L. N., "Solution of the Navier-Stokes Equations for the Flow About a Rotor Blade", Journal of the American Helicopter Society, April 1989.

11. Rai, M. M., "Navier-Stokes Simulations of Rotor-Stator Interaction Using Patched and Overlaid Grids", AIAA Paper 85-1519-CP, July 1985.

12. Gatlin, B. and Whitfield, D. L., "An Implicit Upwind Finite Volume Scheme for Solving the Three-Dimensional Thin-Layer Navier-Stokes Equations", AIAA Paper 87-1149-CP, June 1987.

13. Sankar, L. N. and Kwon, O. J., "Viscous Flow Simulation of Fighter Aircraft", AIAA Paper 91-0278, January 1991.

14. Saad, Y. and Schultz, M.H, "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems", SIAM J. Sci. Stat. Comp., Vol. 7, No. 3, 1986, pp.856-869.

15. Wigton, L. B., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamics Codes", AIAA Paper 85-1494-CP, 1985.

16. Venkatakrishnan, V. and Mavriplis, D. J., "Implicit Solvers for Unstructured Meshes", ICASE Report 91-40, May 1991.

17. Saad, Y. and Semeraro, B. D. , Application of Krylov Exponential Propagation to Fluid Dynamics Equations", AIAA Paper 91-1567-CP, 1991.

18. Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, No. 4, April, 1976.

19. Swanson, R. C. and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations", AIAA Paper 87-1107-CP, June 1987.

Appendix A

Formulation of the ADI Preconditioner

One preconditioner used for the GMRES formulation is a Newton iteration ADI solver. This code is used as a function evaluator for the GMRES, as described in the next section. A brief outline of the Newton algorithm is given below.

i) Discretization in Time and Space

The 3-D Reynolds averaged Navier-Stokes equations written in curvilinear form are given as:

$$\partial_\tau q + \partial_\xi E + \partial_\eta F + \partial_\zeta G = \frac{1}{Re}\left\langle \partial_\xi R + \partial_\eta S + \partial_\zeta T \right\rangle \qquad (A.1)$$

This equation is discretized by using the Euler implicit scheme, which is first order accurate in time and second order accurate in space. The time derivative is approximated by a first order forward difference, while the spatial derivatives are represented by second order central differences. Using Taylor series expansions, Eq. (A.1) can be rewritten:

$$\frac{\left(q^{n+1,k+1} - q^{n+1,k}\right)}{\Delta\tau} + \left(\delta_\xi E^{n+1,k+1} + \delta_\eta F^{n+1,(k,k+1)} + \delta_\zeta G^{n+1,k+1}\right)$$
$$= \frac{-\left(q^{n+1,k} - q^{n}\right)}{\Delta\tau} + \frac{1}{Re}\left(\delta_\xi R^{n+1,k} + \delta_\eta S^{n+1,k} + \delta_\zeta T^{n+1,k}\right)$$
$$+ O\left(\Delta\tau, \Delta\xi^2, \Delta\eta^2, \Delta\zeta^2\right) \qquad (A.2)$$

where $O(\Delta\tau,\Delta\xi^2,\Delta\eta^2,\Delta\zeta^2)$ indicates that this expression is first order accurate in time (second order terms are truncated), and second order accurate in space. In Eq. (A.2), 'n' refers to the time level and 'k' refers to the Newton iteration level at that time step. The notation (k,k+1) will be explained in the next section.

ii) Linearization of the Governing Equation

Given the flow variables at the 'n' time level, equation set (A.2) can now be solved to obtain the flow variables at the 'n+1' time level. Unfortunately, this set of algebraic equations are coupled and highly nonlinear, making them very difficult to solve. To make these equations easier to solve, the convection terms **E** and **G** are linearized about time level 'n+1' and iteration level 'k' by means of Taylor series. When this is substituted into (A.2), the linearized equations are written as:

$$\left\{I + \Delta\tau\delta_\xi \mathbf{A}^{n+1,k} + \Delta\tau\delta_\zeta \mathbf{C}^{n+1,k}\right\}\left\{\Delta \mathbf{q}^{n+1,k+1}\right\} =$$
$$-\Delta\tau\left(\frac{\mathbf{q}^{n+1,k} - \mathbf{q}^{n}}{\Delta\tau}\right) - \Delta\tau\left(\delta_\xi \mathbf{E}^{n+1,k} + \delta_\eta \mathbf{F}^{n+1,(k,k+1)} + \delta_\zeta \mathbf{G}^{n+1,k}\right)$$
$$+\frac{\Delta\tau}{Re}\left(\delta_\xi \mathbf{R}^{n+1,k} + \delta_\eta \mathbf{S}^{n+1,k} + \delta_\zeta \mathbf{T}^{n+1,k}\right) \qquad \text{(A.3)}$$

where

$$\mathbf{A} = \frac{\partial \mathbf{E}}{\partial \mathbf{q}}$$
$$\mathbf{C} = \frac{\partial \mathbf{G}}{\partial \mathbf{q}} \qquad \text{(A.3b)}$$

This equation set is first order accurate in time and second order accurate in space. The matrix to be solved is in block pentadiagonal form.

The solution procedure employs a sweep in the spanwise direction, solving Eq. (A.3) on each spanwise plane. The notation (k,k+1) on the term **F** indicates that the newest available values of the flow variables are to be used in the computation of the residual for each spanwise plane (i.e., the plane on one side will have already been updated).

iii) Approximate Factorization of the Governing Equation

Equation (A.3) is a large, sparse pentadiagonal block matrix equation. This is still very expensive to solve, requiring large amounts of storage and

computation. Instead of solving Eq. (A.3) directly, it is factored into a series of one dimensional block tridiagonal systems of equations, using the approximate factorization technique of Beam and Warming (Ref. 18).

In this method, the left hand side of Eq. (A.3) is approximately factored into two operators:

$$\left\{I + \Delta\tau\delta_\xi A^{n+1,k}\right\}\left\{I + \Delta\tau\delta_\zeta C^{n+1,k}\right\}\left\{\Delta q^{n+1,k}\right\} =$$
$$\Delta\tau\left\{RHS^{n+1,k}\right\} - \Delta\tau^2\delta_\xi A^{n+1,k}\delta_\eta C^{n+1,k}\Delta q^{n+1,k} \qquad (A.4)$$

where

$$\Delta\tau\left\{RHS^{n+1,k}\right\} =$$
$$-\Delta\tau\left(\frac{q^{n+1,k} - q^n}{\Delta\tau}\right) - \Delta\tau\left\{\delta_\xi E^{n+1,k} + \delta_\eta F^{n+1,(k,k+1)} + \delta_\zeta G^{n+1,k}\right\}$$
$$+ \frac{\Delta\tau}{Re}\left\{\delta_\xi R^{n+1,k} + \delta_\eta S^{n+1,k} + \delta_\zeta T^{n+1,k}\right\} \qquad (A.5)$$

The last term on the right hand side of Eq. (A.4) is second order in time, and can thus be dropped without degrading the formal first order time accuracy of the scheme. This gives the factored set of equations to be solved:

$$\left\{I + \Delta\tau\delta_\xi A^{n+1,k}\right\}\left\{I + \Delta\tau\delta_\eta C^{n+1,k}\right\}\left\{\Delta q^{n+1,k}\right\} = \Delta\tau\left\{RHS^{n+1,k}\right\} \qquad (A.6)$$

The solver sweeps in the spanwise direction (η), solving Eq. (A.6) in each spanwise plane. In a spanwise plane, Eq. (A.6) is solved by performing two sweeps. First, a sweep in the ξ direction:

$$\left\{I + \Delta\tau\delta_\xi A^{n+1,k}\right\}\left\{\Delta q^*\right\} = \Delta\tau\left\{RHS^{n+1,k}\right\} \qquad (A.7)$$

where $\{\Delta q^*\}$ is a temporary vector.

The next sweep is in the ζ direction:

$$\left\{I + \Delta\tau\delta_\zeta C^{n+1,k}\right\}\left\{\Delta q^{n+1,k}\right\} = \left\{\Delta q^*\right\} \qquad (A.8)$$

These two sweeps each require the solution of a tridiagonal block matrix, which is computationally more efficient than the solution of the original pentadiagonal block matrix.

Since central differencing is used for the spatial derivatives, each block consists of a 4x4 matrix in 2-D, and a 5x5 matrix in 3-D. Eqs. (A.7) and (A.8) are solved by the block LU decomposition method.

In solving Eq. (A.6) for subsonic and transonic flows, it is necessary to add artificial viscosity to damp the numerical oscillations. The numerical viscosity model proposed by Jameson, Turkel, and Schmidt, and modified by Swanson and Turkel (Ref. 19) is used. On the left side, an implicit smoothing was also added. Equations (A.7) and (A.8) then become:

$$\left\{ I + \Delta\tau\delta_\xi A^{n+1,k} - \frac{\varepsilon_I}{J}\Delta t\, \delta_{\xi\xi} J \right\} \left\{ \Delta q^* \right\} = \Delta\tau \left\{ RHS^{n+1,k} \right\} \qquad \text{(A.9)}$$

and

$$\left\{ I + \Delta\tau\delta_\zeta C^{n+1,k} - \frac{\varepsilon_I}{J}\Delta t\, \delta_{\zeta\zeta} J \right\} \left\{ \Delta q^{n+1,k} \right\} = \left\{ \Delta q^* \right\} \qquad \text{(A.10)}$$

When viscous flows at high Reynolds numbers are solved, it becomes necessary to consider turbulent effects. While the present equations can directly model turbulent motion, the small time step and dense grid that is required make the computational cost prohibitive. To keep a reasonable grid spacing, Eq. (A.3) is time-averaged and the well-known Baldwin-Lomax turbulence model is employed to represent the turbulent stresses.

Appendix B

Formulation of the LU-SGS Preconditioner

The discretized 3-D Navier-Stokes equations in curvilinear coordinates are written:

$$\left[I + \Delta\tau\delta_\xi \mathbf{A}^{n+1,k} + \Delta\tau\delta_\eta \mathbf{B}^{n+1,k} + \Delta\tau\delta_\zeta \mathbf{C}^{n+1,k}\right]\left\{\Delta q^{n+1,k+1}\right\} = \Delta\tau\left[RHS^{n+1,k}\right] \tag{B.1}$$

where

$$\Delta\tau\left[RHS^{n+1,k}\right] = -\Delta\tau\left(\frac{q^{n+1,k} - q^{n}}{\Delta\tau}\right) - \Delta\tau\left(\delta_\xi E^{n+1,k} + \delta_\eta F^{n+1,(k,k+1)} + \delta_\zeta G^{n+1,k}\right) + \frac{\Delta\tau}{Re}\left(\delta_\xi R^{n+1,k} + \delta_\eta S^{n+1,k} + \delta_\zeta T^{n+1,k}\right) \tag{B.2}$$

and

$$\mathbf{A} = \frac{\partial E}{\partial q}$$
$$\mathbf{B} = \frac{\partial F}{\partial q}$$
$$\mathbf{C} = \frac{\partial G}{\partial q} \tag{B.3}$$

An LU decomposition can be used to rewrite Eq (B.1) as:

$$\left[I + \Delta t\left(D_\xi^+ \mathbf{A}^- + D_\xi^- \mathbf{A}^+ + D_\eta^+ \mathbf{B}^- + D_\eta^- \mathbf{B}^+ + D_\zeta^+ \mathbf{C}^- + D_\zeta^- \mathbf{C}^+\right)^{n+1,k}\right]\left\{\Delta q^{n+1,k+1}\right\} = \Delta\tau\left[RHS^{n+1,k}\right] \tag{B.4}$$

where

$$\mathbf{A}^{+} = \frac{1}{2}\left(\mathbf{A} + \beta\lambda_A \mathbf{I}\right)$$

$$\mathbf{A}^{+} = \frac{1}{2}\left(\mathbf{A} - \beta\lambda_A \mathbf{I}\right)$$

$$\mathbf{B}^{+} = \frac{1}{2}\left(\mathbf{B} + \beta\lambda_B \mathbf{I}\right)$$

$$\mathbf{B}^{-} = \frac{1}{2}\left(\mathbf{B} - \beta\lambda_B \mathbf{I}\right)$$

$$\mathbf{C}^{+} = \frac{1}{2}\left(\mathbf{C} + \beta\lambda_C \mathbf{I}\right)$$

$$\mathbf{C}^{-} = \frac{1}{2}\left(\mathbf{C} - \beta\lambda_C \mathbf{I}\right) \qquad \text{(B.5)}$$

β is a user defined scaling factor (1.2 is used at present) used to adjust the magnitude of the main diagonals, and

$$\lambda_A = \left| |U| + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2} \right|$$

$$\lambda_B = \left| |V| + a\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2} \right|$$

$$\lambda_C = \left| |W| + a\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2} \right| \qquad \text{(B.6)}$$

where **U**, **V**, and **W** are the contravariant velocities. Note that the right hand side residual is the same as that for the ADI preconditioner; in fact, the same subroutines are used to compute the RHS.

The derivatives are given as:

$$D_\xi^{+} = D_{i+1} - D_i$$

$$D_\xi^{-} = D_i - D_{i-1} \qquad \text{(B.7)}$$

At this point, Eq (2) is rewritten in nonconservative form:

$$\left\{ I + \Delta t\left(A^{-}D_{\xi}^{+} + A^{+}D_{\xi}^{-} + B^{-}D_{\eta}^{+} + B^{+}D_{\eta}^{-} + C^{-}D_{\zeta}^{+} + C^{+}D_{\zeta}^{-} \right)^{n+1,k} \right\} \left\{ \Delta q^{n+1,k+1} \right\} =$$

$$\Delta\tau \left\{ RHS^{n+1,k} \right\} \qquad \text{(B.8)}$$

The nonconservative form reduces the memory necessary for the LU solver.

Discretization of Eq. (B.8) yields a sparse matrix with 7 diagonals. After dividing Eq (B.8) into lower and upper matrices, Eq (B.8) can be solved by a two step method:

$$\left\{ I + \Delta t\left(A^{-}D_{\xi}^{+} + B^{-}D_{\eta}^{+} + C^{-}D_{\zeta}^{+} + A^{+} + B^{+} + C^{+} \right)^{n+1,k} \right\} \left\{ \Delta q^{*} \right\} =$$

$$\Delta\tau \left\{ RHS^{n+1,k} \right\} \qquad \text{(B.9)}$$

$$\left\{ I + \Delta t\left(A^{+}D_{\xi}^{-} + B^{+}D_{\eta}^{-} + C^{+}D_{\zeta}^{-} - A^{-} - B^{-} - C^{-} \right)^{n+1,k} \right\} \left\{ \Delta q^{n+1,k+1} \right\} =$$

$$\left\{ I + \Delta t\left(A^{+} + B^{+} + C^{+} - A^{-} - B^{-} - C^{-} \right)^{n+1,k} \right\} \left\{ \Delta q^{*} \right\} \qquad \text{(B.10)}$$

With this method, no matrix inversions are required; at each step in each sweep, everything but the main diagonal is known and moved to the right hand side. Memory is greatly lessened, and no implicit dissipation is necessary on the left hand side of the equation.

Appendix C

Formulation of the GMRES solver

The iterative ADI and LU formulations may be expressed in this way:

$$\mathbf{q}^{n+1,k+1} = \mathbf{F}(\mathbf{q}^{n+1,k}) \qquad \text{(C.1)}$$

In words, given a guess for $\mathbf{q}^{n+1,k}$, the solver returns a (hopefully) better approximation to the correct solution $\mathbf{q}^{n+1,k+1}$. When the solution has converged (i.e., $\mathbf{q}^{n+1,k} = \mathbf{q}^{n+1,k+1}$), then:

$$\mathbf{q}^{n+1,k} - \mathbf{F}(\mathbf{q}^{n+1,k}) = \mathbf{M}(\mathbf{q}^{n+1,k}) = 0 \qquad \text{(C.2)}$$

The GMRES solver uses the original iterative ADI or LU solver as a function evaluator (i.e., given a set of input flow properties, the solver sends back an updated set of flow properties), and computes the set of flow properties that will satisfy Eq. (C.2).

The GMRES solver starts by finding a set of orthonormal direction vectors which define a subspace of the total space spanned by the problem. Once this subspace is defined, the error magnitude is projected upon it. From here, a least squares problem is solved to reduce the error as much as possible in the subspace.

Obviously, the success and speed of the GMRES solution method depends greatly on the original flow solver's ability to help define useful direction vectors, and hence a subspace that contains much of the error components. This is why both the ADI and LU formulations are being investigated.

The J direction vectors are found as follows:

First, the initial direction is computed as

$$\vec{d}_1 = \mathbf{M}(\mathbf{q}^{n+1,k}) \qquad \text{(C.3)}$$

and normalized as

$$\vec{d}_1 = \frac{\vec{d}_1}{\|\vec{d}_1\|} \qquad \text{(C.4)}$$

To compute the remaining search directions (j=1,2,..,J-1), take

$$\vec{d}_{j+1} = \overline{M}(q^{n+1,k}\,;\,\vec{d}_j) - \sum_{i=1}^{j} b_{ij}\,\vec{d}_i \qquad \text{(C.5)}$$

where

$$b_{ij} = (\overline{M}(q^{n+1,k}\,;\,\vec{d}_j),\,\vec{d}_i) \qquad \text{(C.6)}$$

and

$$\overline{M}(q\,;\,\vec{d}) = \frac{M(q + \varepsilon\vec{d}) - M(q)}{\varepsilon} \qquad \text{(C.7)}$$

Here, e is taken to be some small number. In this work, e is taken to be 0.001.

The new direction $\vec{d}_{j+1}$ is normalized before the next direction is computed:

$$b_{j+1,j} = \|\vec{d}_{j+1}\|, \qquad \text{(C.8)}$$

and

$$\vec{d}_{j+1} = \frac{\vec{d}_{j+1}}{b_{j+1,j}} \qquad \text{(C.9)}$$

After obtaining the search directions, the solution vector is updated using

$$q^{n+1,k+1} = q^{n+1,k} + \sum_{j=1}^{J} a_j\vec{d}_j \qquad \text{(C.10)}$$

where the coefficients a_j are chosen to minimize:

$$\left\| \mathbf{M}(q^{n+1,k+1}) \right\|^2 = \left\| \mathbf{M}(q^{n+1,k} + \sum_{j=1}^{J} a_j \vec{d}_j) \right\|^2$$

$$\cong \left\| \mathbf{M}(q^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{\mathbf{M}}(q^{n+1,k}; \vec{d}_j) \right\|^2 . \qquad \text{(C.11)}$$

This equation is minimized as follows:

Let D_j be the matrix of directions $\{d_1, d_2, d_3, ..., d_j\}$.. Also, let F_j be the matrix of directional derivatives given as $\{M_1, M_2, M_3, ..., M_j\}$, where:

$$M_j = \overline{M}(q^{n+1,k}; \vec{d}_j) \qquad \text{(C.12)}$$

Then Eq. (C.5) may be rewritten in matrix form as:

$$M_j = D_{j+1} B \qquad \text{(C.13)}$$

Here, B is the (J+1) x (J) matrix:

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} & & & b_{1,J-2} & b_{1,J-1} & b_{1,J} \\ b_{2,1} & b_{2,2} & b_{2,3} & & \cdots & b_{2,J-2} & b_{2,J-1} & b_{2,J} \\ 0 & b_{3,2} & b_{3,3} & & & b_{3,J-2} & b_{3,J-1} & b_{3,J} \\ & 0 & & & \ddots & & \vdots & \\ & & & & & b_{J-1,J-2} & b_{J-1,J-1} & b_{J-1,J} \\ & 0 & & & 0 & 0 & b_{J,J-1} & b_{J,J} \\ & & & & & 0 & 0 & b_{J+1,J} \end{bmatrix} \qquad \text{(C.14)}$$

Note that at this point, $b_{J+1,J}$ is not yet known. Saad and Schultz give the following formula for evaluating this term without another function evaluation:

$$b_{J+1,J}^2 = \left\| \overline{\mathbf{M}}(q^{n+1,k}; \vec{d}_J) \right\|^2 - \sum_{i=1}^{J} b_{i,J}^2 \qquad \text{(C.15)}$$

At this point, Eq. (C.11) is rewritten:

$$\left\| M(q^{n+1,k}) + \sum_{j=1}^{J} a_j \overline{M}(q^{n+1,k}; \vec{d}_j) \right\|^2$$
$$= \left\| M(q^{n+1,k}) + M_j A \right\|^2 \qquad \text{(C.16)}$$

where A is the vector $\{a_1, a_2, a_3, ..., a_j\}^T$. Then, using the definition of the first direction and Eq. (C.13), Eq. (C.16) becomes:

$$\left\| M(q^{n+1,k}) + M_j A \right\|^2$$
$$= \left\| \left(\left\| \vec{d}_1 \right\| \vec{d}_1 + M_j A \right) \right\|^2$$
$$= \left\| \left(\left\| \vec{d}_1 \right\| \vec{d}_1 + D_{j+1} B A \right) \right\|^2$$
$$= \left\| D_{j+1} \left(\left\| \vec{d}_1 \right\| e + BA \right) \right\|^2$$
$$= \left\| \left(\left\| \vec{d}_1 \right\| e + BA \right) \right\|^2 \qquad \text{(C.17)}$$

where e is the first column of the (JxJ) identity matrix.

This least squares problem is solved using the QR algorithm in LINPACK.

Figure 1: Comparison of 5 Direction GMRES to Hybrid ADI Solver

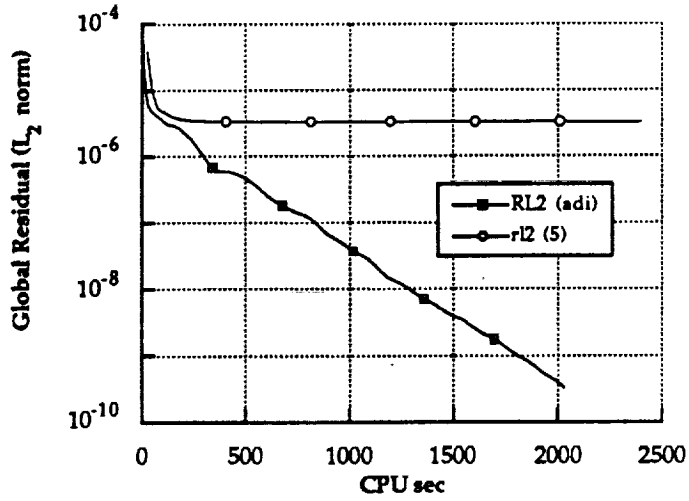


Figure 2: Pressure Coefficient Comparison GMRES (5) vs. Hybrid ADI

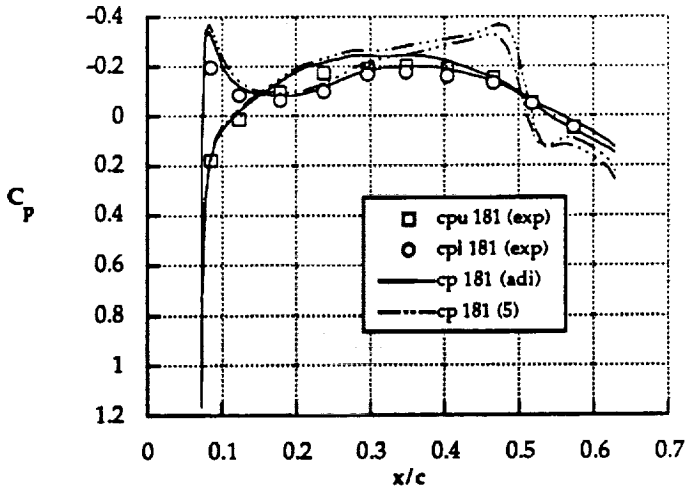
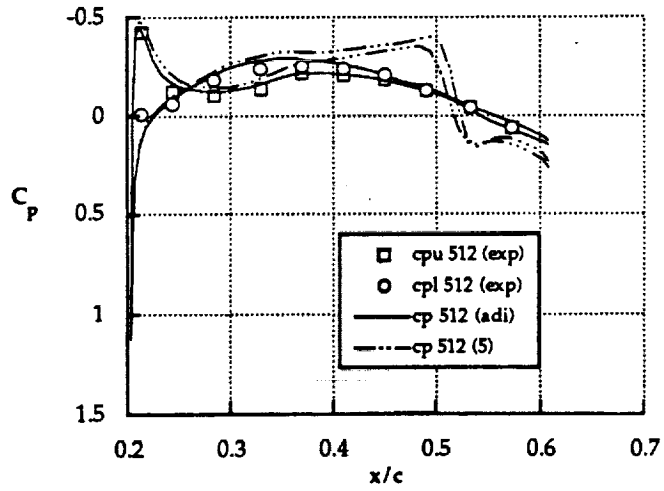
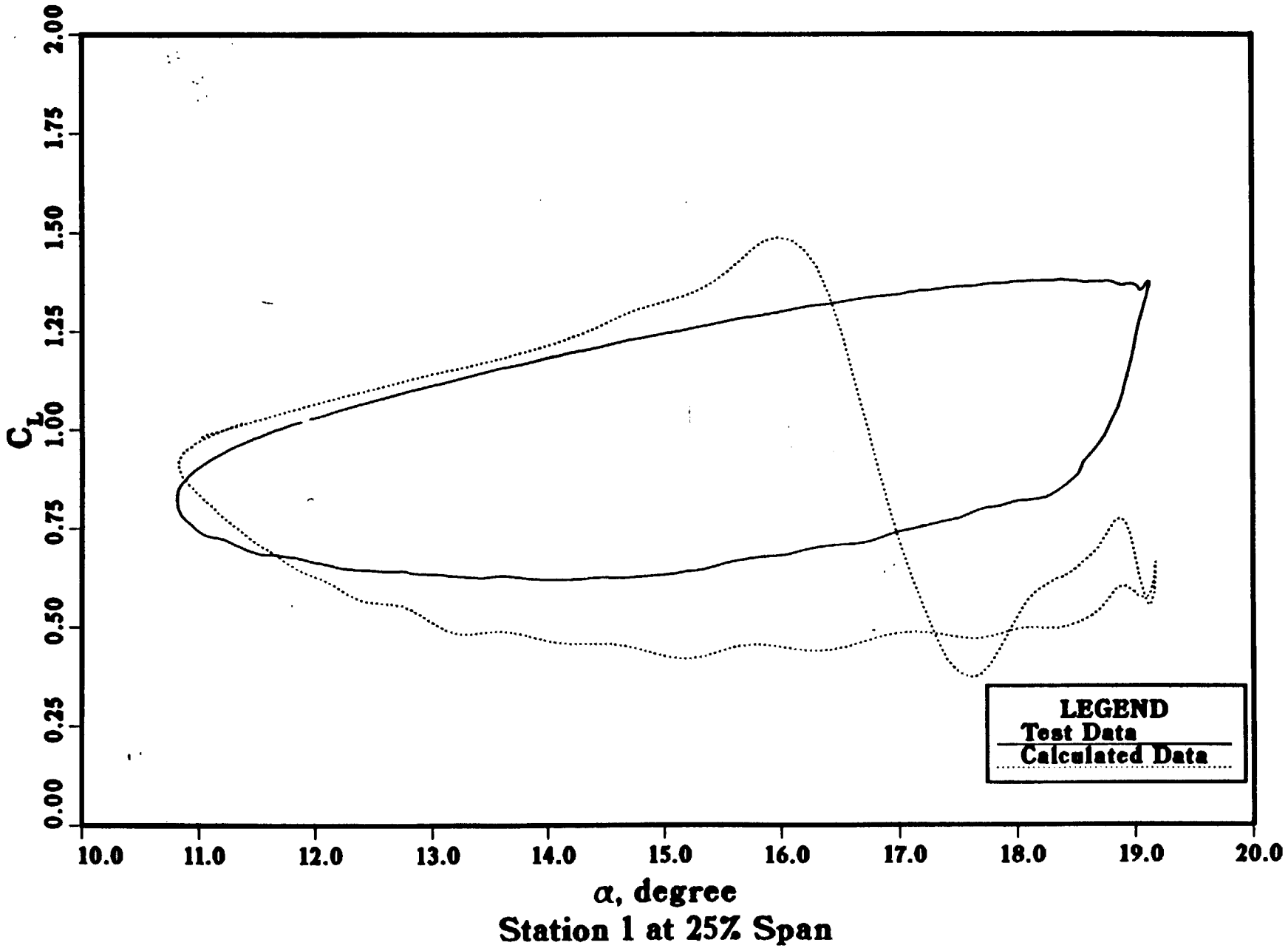


Figure 3: Pressure Coefficient Comparison GMRES (5) vs. Hybrid ADI



Lift Coefficient vs Alpha

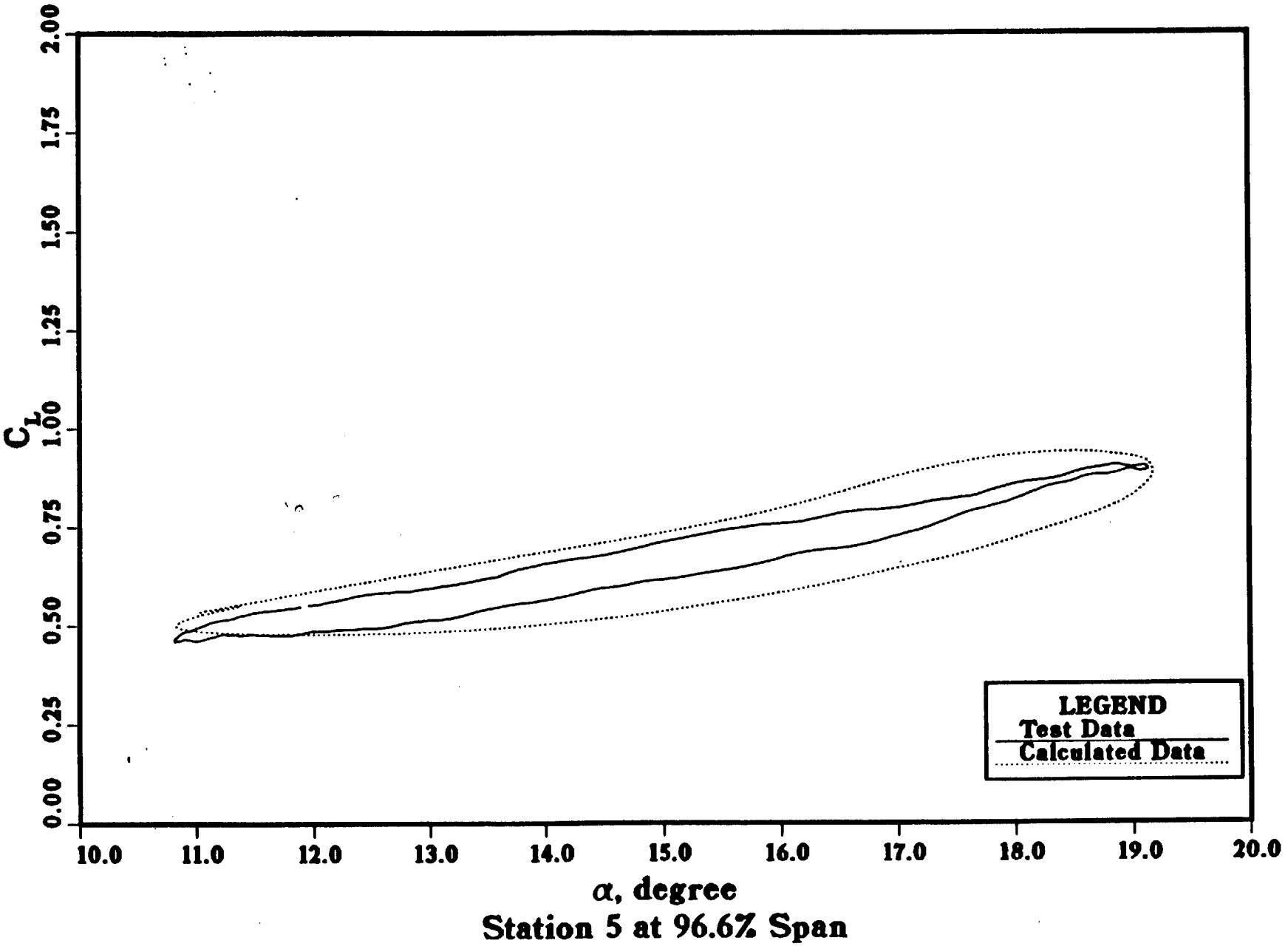
Alpha = 15 +/− 4 deg, Freq = 10 Hz



Station 1 at 25% Span

Lift Coefficient vs Alpha

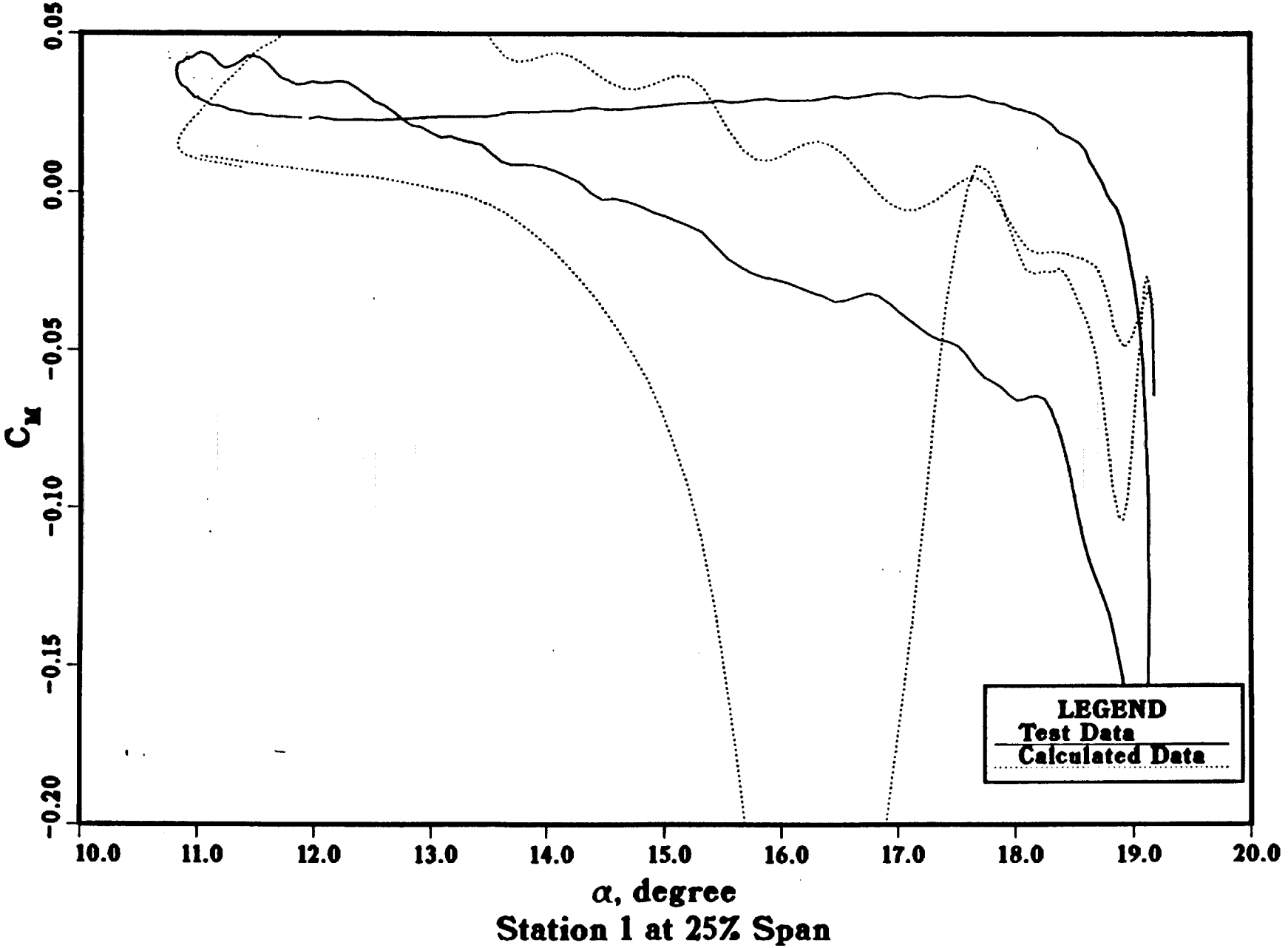
Alpha = 15 +/− 4 deg, Freq = 10 Hz



Station 5 at 96.6% Span

Pitch Moment Coefficient vs Alpha

Alpha = 15 +/− 4 deg, Freq = 10 Hz

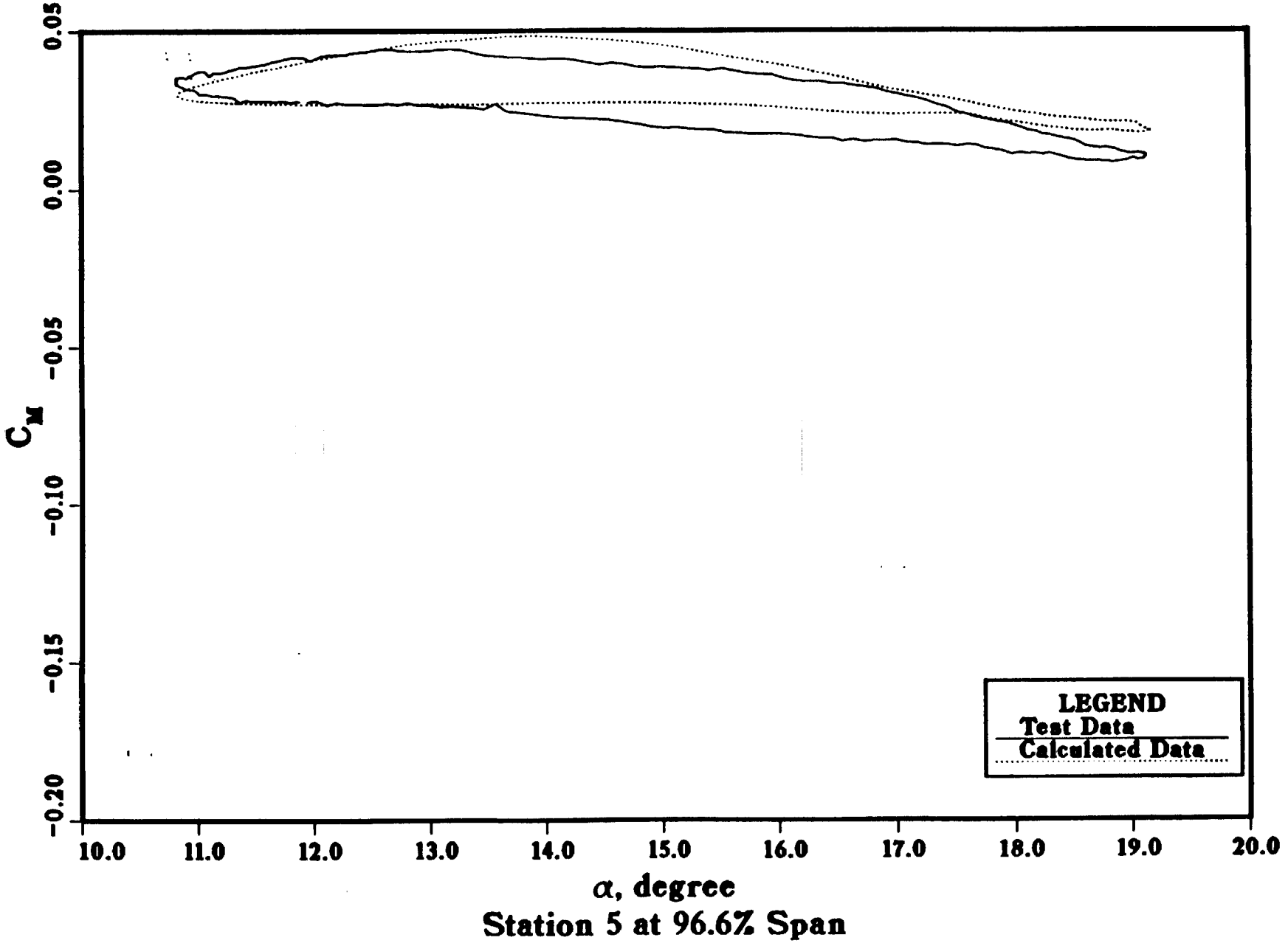


α, degree

Station 1 at 25% Span

Pitch Moment Coefficient vs Alpha

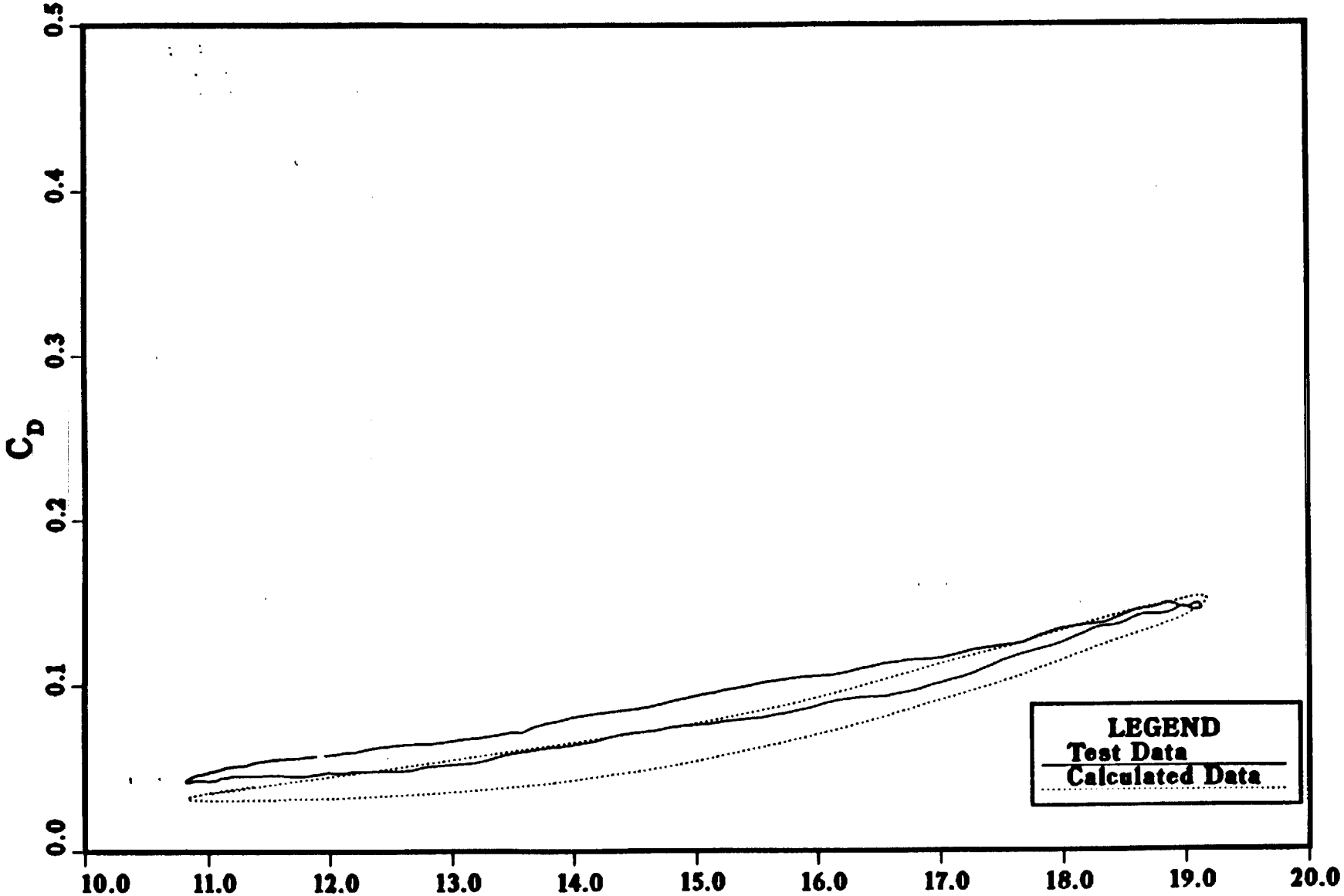
Alpha = 15 +/− 4 deg, Freq = 10 Hz



Station 5 at 96.6% Span

Drag Coefficient vs Alpha

Alpha = 15 +/− 4 deg, Freq = 10 Hz



Station 5 at 96.6% Span

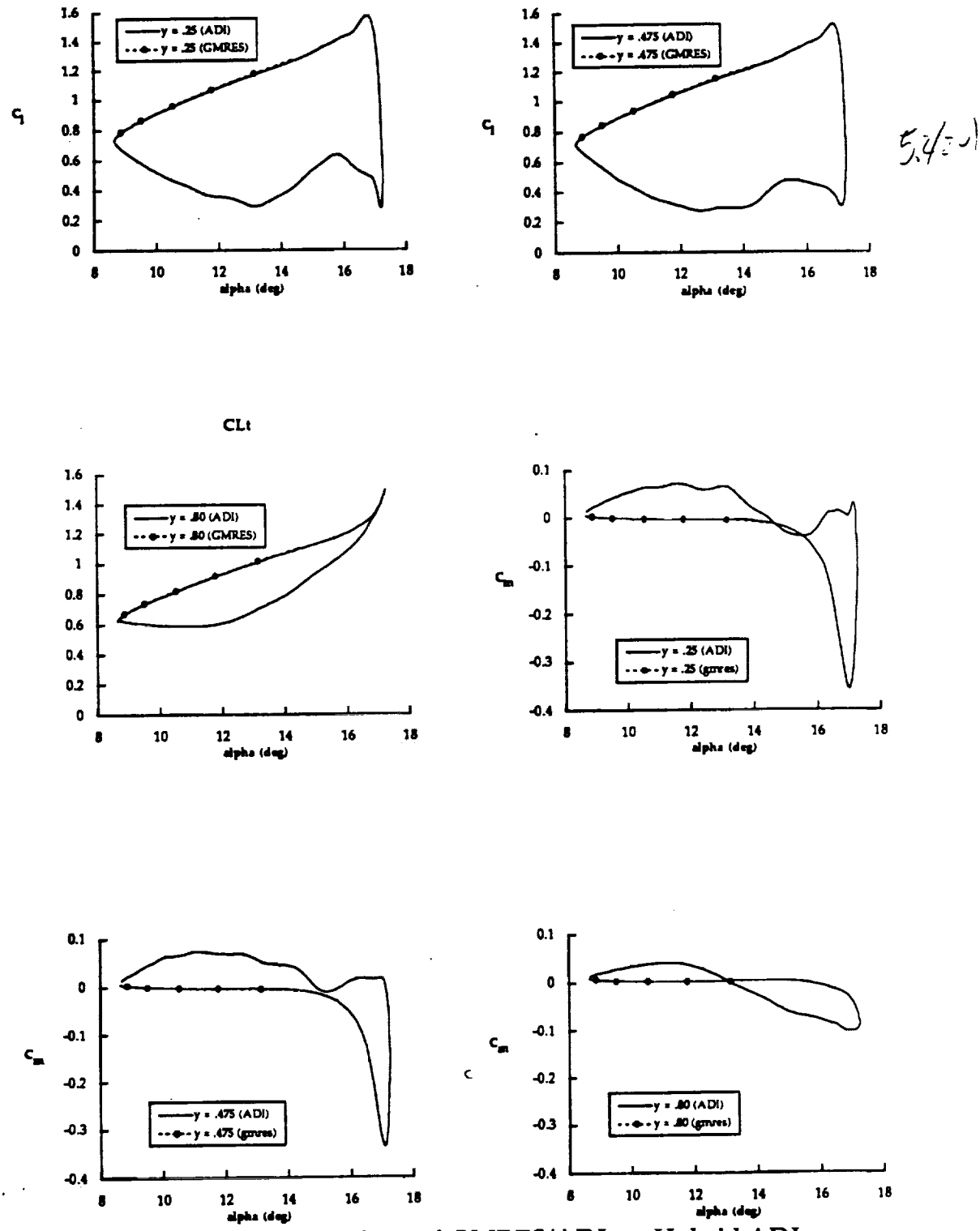


Figure 9: Preliminary Comparison of GMRES/ADI vs. Hybrid ADI for a NACA 0015 Wing (AR = 5) in Dynamic Stall (α_{mean} = 13°; α_{pitch} = 4°; f = 14 Hz)

Figure 10: Global Residual Comparison
LU-SGS vs. Hybrid ADI

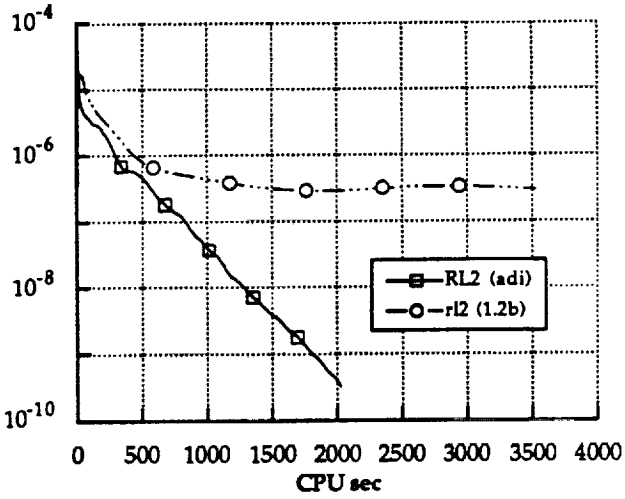


Figure 11: Pressure Coefficient Comparison
LU-SGS solver vs. Experiment

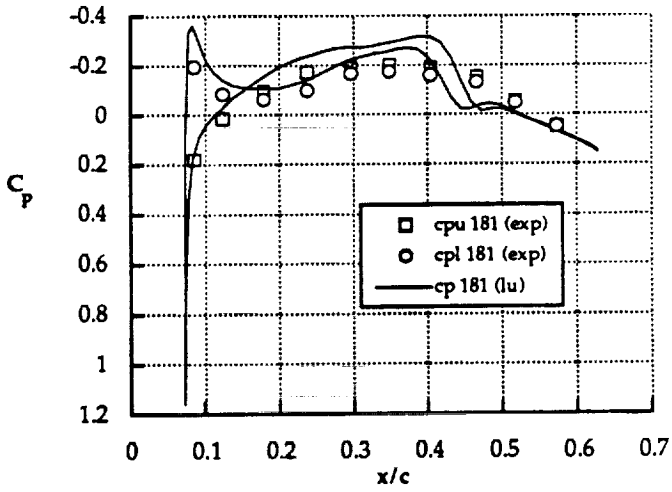


Figure 12: Pressure Coefficient Comparison
LU-SGS Solver vs. Experiment

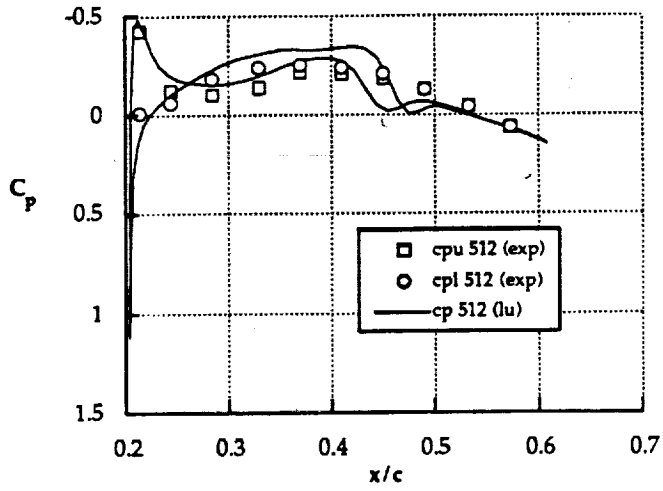


Figure 13: Global Residual Comparison
GMRES (20-LU) vs. LU-SGS Solver

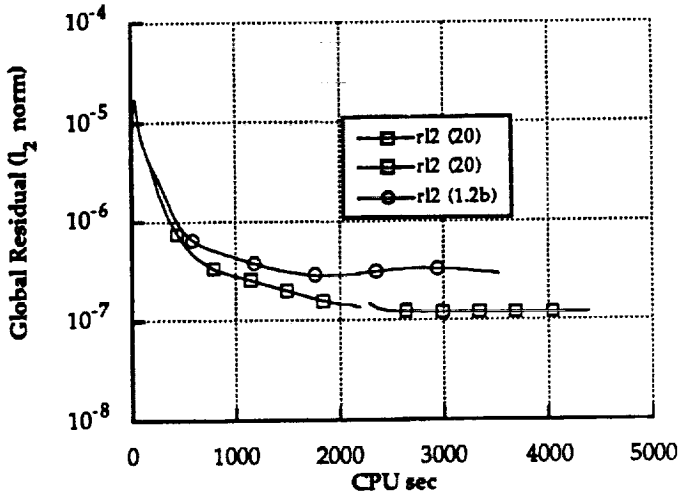


Figure 14: Pressure Coefficient Comparison
GMRES (20-LU) vs. LU-SGS Solver

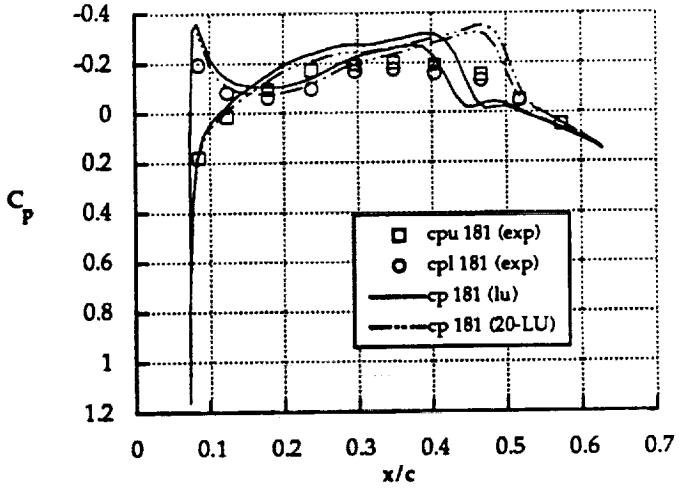


Figure 15: Pressure Coefficient Comparison
GMRES (20-LU) vs. LU-SGS Solver

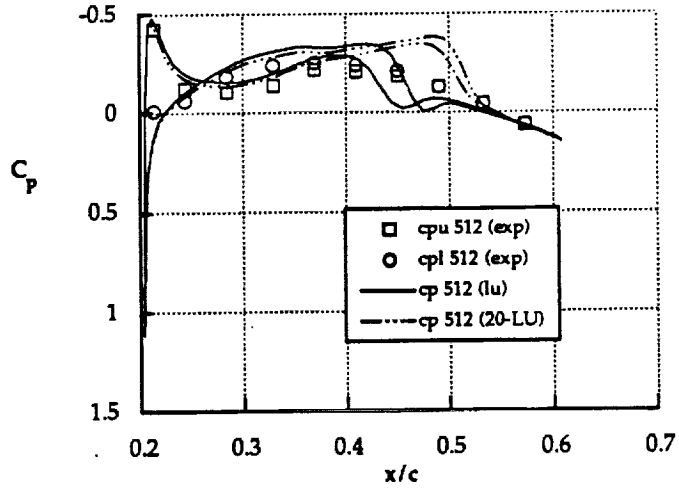


Figure 16: Global Residual Comparison
GMRES (20-20i) vs. GMRES (5-10i) vs. Hybrid ADI

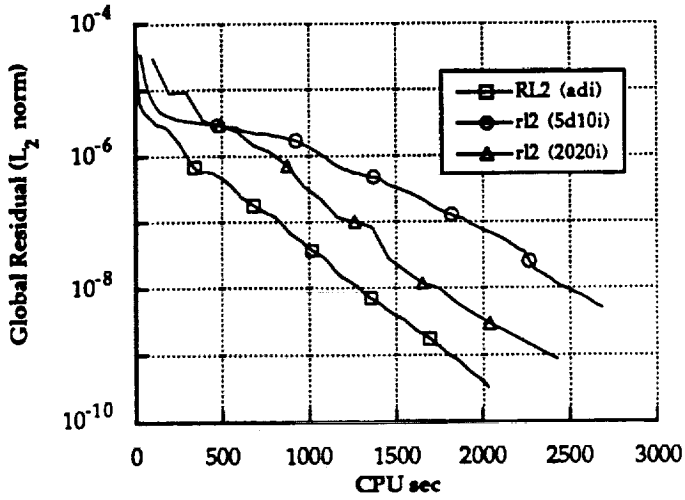


Figure 17: Pressure Coefficient Comparison
GMRES (20-20i) vs. ADI Solver

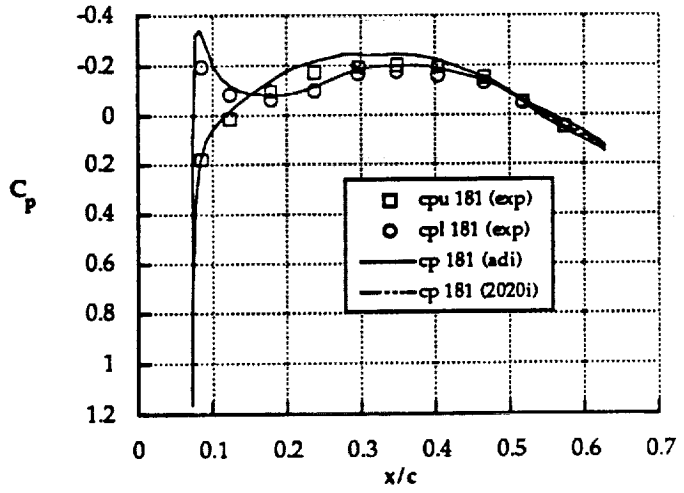


Figure 18: Pressure Coefficient Comparison
GMRES (20-20i) vs. ADI Solver